SPABERT: A Pretrained Language Model from Geographic Data for Geo-Entity Representation

Zekun Li1, Jina Kim1, Yao-Yi Chiang1, Muhao Chen2
1Department of Computer Science and Engineering, University of Minnesota, Twin Cities
2Department of Computer Science, University of Southern California
{li002666,kim01479,yaoyi}@umn.edu, muhaoche@usc.edu

Abstract

Named geographic entities (geo-entities for short) are the building blocks of many geographic datasets. Characterizing geo-entities is integral to various application domains, such as geo-intelligence and map comprehension, while a key challenge is to capture the spatial-varying context of an entity. We hypothesize that we shall know the characteristics of a geo-entity by its surrounding entities, similar to knowing word meanings by their linguistic context. Accordingly, we propose a novel spatial language model, SPABERT, which provides a general-purpose geo-entity representation based on neighboring entities in geospatial data. SPABERT extends BERT to capture linearized spatial context, while incorporating a spatial coordinate embedding mechanism to preserve spatial relations of entities in the 2-dimensional space. SPABERT is pre-trained with masked language modeling and masked entity prediction tasks to learn spatial dependencies. We apply SPABERT to two downstream tasks: geo-entity typing and geo-entity linking. Compared with the existing language models that do not use spatial context, SPABERT shows significant performance improvement on both tasks. We also analyze the entity representation from SPABERT in various settings and the effect of spatial coordinate embedding.

1 Introduction

Interpreting human behaviors requires considering human activities and their surrounding environment. Looking at a stopping location, [Speedway, ¤],1 from a person’s trajectory, we might assume that this person needs to use the location’s amenities if [Speedway] implies a gas station, and ¤ is near a highway exit. We might predict a meetup at [Speedway, ¤] if the trajectory travels through many other locations, 9, 9, ..., of the same name, [Speedway], to arrive at [[Speedway], 9] in the middle of farmlands. As humans, we are able to make such inferences using the name of a geographic entity (geo-entity) and other entities in a spatial neighborhood. Specifically, we contextualize a geo-entity by a reasonable surrounding neighborhood learned from experience and, from the neighborhood, relate other relevant geo-entities based on their name and spatial relations (e.g., distance) to the geo-entity. This way, even if two gas stations have the same name (e.g., ‘Speedway’) and entity type (e.g., ‘gas station’), we can still reason about their spatially varying semantics and use the semantics for prediction.

Capturing this spatially varying location semantics can help recognizing and resolving geospatial concepts (e.g., toponym detection, typing and linking) and the grounding of geo-entities in documents, scanned historical maps, and a variety of knowledge bases, such as Wikidata, OpenStreetMap, and GeoNames. Also, the location semantics can support effective use of spatial textual information (geo-entities names) in many spatial computing tasks, including moving behavior detection from visiting locations of trajectories (Yue et al., 2021, 2019), point of interest recommendations (Yin et al., 2017; Zhao et al., 2022), air quality (Lin et al., 2017, 2018, 2020; Jiang et al., 2019) and traffic prediction (Yuan and Li, 2021; Gao et al., 2019) using location context.

Recently, the research community has seen a rapid advancement in pretrained language models (PLMs) (Devlin et al., 2019; Liu et al., 2019; Lewis et al., 2020; Sanh et al., 2019), which supports strong contextualized language representation abilities (Lan et al., 2020) and serves as the backbones of various NLP systems (Rothe et al., 2020; Yang et al., 2019). The extensions of these PLMs help NL tasks in different data domains (e.g., biomedicine (Lee et al., 2020; Phan et al., 2021), software engineering (Tabassum et al., 2020), fi-
Figure 1: Overview for generating the pivot entity representation, $E_p$ (pivot entity; $\bullet$: neighboring geo-entities).

SpABERT sorts and concatenate neighboring geo-entities by their distance to pivot in ascending order to form a pseudo-sentence. [$CLS$] is prepended at the beginning. [$SEP$] separates entities. SpABERT generates token representations and aggregates representations of pivot tokens to produce $E_p$.

To tackle these challenges, we present SpABERT ($\bullet$), a LM that captures the spatially varying semantics of geo-entity names using large geographic datasets for entity representation. Built upon BERT (Devlin et al., 2019), SpABERT generates the contextualized representation for a geo-entity of interest (referred to as the pivot), based on its geographically nearby geo-entities. Specifically, SpABERT linearizes the 2-dimensional spatial context by forming pseudo sentences that consist of names of the pivot and neighboring geo-entities, ordered by their spatial distance to the pivot. SpABERT also encodes the spatial relations between the pivot and neighboring geo-entities with a continuous spatial coordinate embedding. The spatial coordinate embedding models horizontal and vertical distance relations separately and, in turn, can capture the orientation relations. These techniques make the inputs compatible with the BERT-family structures.

In addition, the backbone LM, BERT, in SpABERT is pretrained with general-purpose corpora and would not work well directly on geo-entity names because of the domain shift. We thus train SpABERT using pseudo sentences generated from large geographic datasets derived from OpenStreetMap (OSM). This pretraining process conducts Masked Language Modeling (MLM) and Masked Entity Prediction (MEP) that randomly masks subtokens and full geo-entity names in the pseudo sentences, respectively. MLM and MEP enable SpABERT to learn from pseudo sentences for generating spatially varying contextualized representations for geo-entities.

SpABERT provides a general-purpose representation for geo-entities based on their spatial context. Similar to linguistic context, spatial context refers to the surrounding environment of a geo-entity. For example, [[Speedway], [ ], [Speedway], ] and [Speedway], [ ] would have different representations since the surrounding environment of [ ], [ ], and [ ] could vary. We evaluate SpABERT on two tasks: 1) geo-entity typing and 2) geo-entity linking to external knowledge bases. Our analysis includes the performance comparison of SpABERT in various settings due to characteristics of geographic data sources (e.g., entity omission).

To summarize, this work has the following contributions. We propose an approach to linearize the 2-dimensional spatial context, encode the geo-entity spatial relations, and use a LM to produce spatially varying feature representations of geo-entities. We show that SpABERT is a general-purpose encoder by supporting geo-entity typing and geo-entity linking, which are keys to effectively integrating large varieties of geographic data sources, the grounding of geo-entities as well as supports for a broad range of spatial computing applications. The experiments demonstrate that SpABERT is effective for both tasks and outperforms the SOTA LMs.

2OpenStreetMap: https://www.openstreetmap.org/
2 SpABERT

SpABERT is a LM built upon a pretrained BERT and further trained to produce contextualized geo-entity representations given large geographic datasets. Fig. 1 shows the outline of SpABERT, with the details below. This section first presents the preliminary (§2.1) and then describes the overall approach for learning geo-entities’ contextualized representations (§2.2), the pretraining strategies (§2.3), and inference procedures (§2.4).

2.1 Preliminary

We assume that given a geographic dataset (e.g., OSM) with many geo-entities, \( S = \{g_1, g_2, ..., g_i\} \), each geo-entity \( g_i \) (e.g., \{Speedway, Bloom Island Park\}) has two attributes: name \( g^{\text{name}} \) (Speedway) and location \( g^{\text{loc}} \). The location attribute, \( g^{\text{loc}} \), is a tuple \( g^{\text{loc}} = (g^{\text{locx}}, g^{\text{locy}}, ...) \) that identifies the location in a coordinate system (e.g., x and y image pixel coordinates or latitude and longitude geo-coordinates with altitudes). WLOG, here we assume a 2-dimensional space. SpABERT aims to generate a contextualized representation for each geo-entity \( g_i \) in \( S \) given its spatial context. We denote a geo-entity that we seek to contextualize as the pivot entity, \( g_{p} \), \( p \) for short. The spatial context of \( p \) is \( \text{SC}(p) = \{g_{n_1}, ..., g_{n_k}\} \) where \( \text{distance}(p, g_{n_k}) < T \). \( T \) is a spatial distance parameter defining a local area. We call \( g_{n_1}, ..., g_{n_k} \) as \( p \)’s neighboring geo-entities and denote them as \( n_1, ..., n_k \) when there is no confusion.

2.2 Contextualizing Geo-entities

Linearizing Neighboring Geo-entity Names For a pivot, \( p \), SpABERT first linearizes its neighboring geo-entity names to form a BERT-compatible input sequence, called a pseudo sentence. The corresponding pseudo sentence for the example in Fig. 1 is constructed as:

\[
\]

The pseudo sentence starts with the pivot name followed by the names of the pivot’s neighboring geo-entities, ordered by their spatial distance to the pivot in ascending order. The idea is that nearby geo-entities are more related (for contextualization) than distant geo-entities. SpABERT also tokenizes the pseudo sentences using the original BERT tokenizer with the special token [SEP] to separate entity names. The subtokens of a neighbor \( n_k \) is denoted as \( T^{n_k} \) as in Fig. 2.

Encoding Spatial Relations SpABERT adopts two types of position embeddings in addition to the token embeddings in the pseudo sentences (Fig. 2). The sequence position embedding represents the token order, same as the original position embedding in BERT and other Transformer LMs. Also, SpABERT incorporates a spatial coordinate embedding mechanism, which seeks to represent the spatial relations between the pivot and its neighboring geo-entities. SpABERT’s spatial coordinate embeddings encode each location dimension separately to capture the relative distance and orientation between geo-entities. Specifically, for the 2-dimensional space, SpABERT generates normalized distance \( \text{dist}^{n_k}_x \) and \( \text{dist}^{n_k}_y \) for each neighboring geo-entity using the following equations:

\[
\text{dist}^{n_k}_x = (g^{\text{locx}}_{n_k} - g^{\text{locx}}_p)/Z \\
\text{dist}^{n_k}_y = (g^{\text{locy}}_{n_k} - g^{\text{locy}}_p)/Z
\]

where \( Z \) is a normalization factor, and \( (g^{\text{locx}}_p, g^{\text{locy}}_p) \), \( (g^{\text{locx}}_{n_k}, g^{\text{locy}}_{n_k}) \) are the locations of pivot \( p \) and neighboring entity \( n_k \). Note that the tokens belong to the same geo-entity name have the same \( \text{dist}^{n_k}_x \) and \( \text{dist}^{n_k}_y \). Also, SpABERT uses DSEP, a constant numerical value larger than \( \max(\text{dist}^{n_k}_x, \text{dist}^{n_k}_y) \) for all neighboring entities to differentiate special tokens from entity name tokens.

SpABERT encodes \( \text{dist}^{n_k}_x \) and \( \text{dist}^{n_k}_y \) using a continuous spatial coordinate embedding layer with real-valued distances as input to preserve the continuity of the output embeddings. Let \( S_{n_k} \) be the spatial coordinate embedding of the neighbor entity \( n_k \), \( M \) be the embedding’s dimension, and \( S_{n_k} \in \mathbb{R}^M \). We define \( S_{n_k} \) as:

\[
S^{(m)}_{n_k} = \begin{cases} 
\sin(\text{dist}^{n_k}/10000^{2j/M}), m = 2j \\
\cos(\text{dist}^{n_k}/10000^{2j/M}), m = 2j + 1
\end{cases}
\]

where \( S^{(m)}_{n_k} \) is the \( m \)-th component of \( S_{n_k} \). \( \text{dist}^{n_k} \) represents \( \text{dist}^{n_k}_x \) and \( \text{dist}^{n_k}_y \) for the spatial coordinate embeddings along the horizontal and vertical directions, respectively.

The token embedding, sequence position embedding and spatial coordinate embedding are summed up then fed into the encoder. Similar to BERT, SpABERT encoder calculates an output embedding for each token. Then SpABERT averages the
pivot’s token-level embeddings to produce a fixed-length embedding for the pivot’s contextualized representation.

2.3 Pretraining

We train SPABERT with two tasks to adapt the pre-trained BERT backbone to geo-entity pseudo sentences. One task is the masked language modeling (MLM) (Devlin et al., 2019), for which SPABERT needs to learn how to complete the full names of geo-entities from pseudo sentences with randomly masked subtokens using the remaining subtokens and their spatial coordinates (i.e., partial names and spatial relations between subtokens). The block below shows an example of the masked input for MLM, where ### are the masked subtokens.

In addition, we hypothesize that given common spatial co-occurrence patterns in the real world, one can use neighboring geo-entities to predict the name of a geo-entity. Therefore, we propose and incorporate a masked entity prediction (MEP) task, which randomly masks all subtokens of an entity name in a pseudo sentence. For MEP, SPABERT relies on the masked entity’s spatial relation to neighboring entities to recover the masked name. The block below shows an example of the masked input for MEP.

Both MLM and MEP have a masking rate of 15% (without masking [CLS] and [SEP]). The sequence position and spatial coordinates are not masked.

Pretraining Data

We construct a large training set from OSM covering the City of London and the State of California. Since OSM is crowd-sourced, we clean the raw data by removing non-alphanumeric place names and geo-entities that do not have a place name or geocoordinates. We randomly select geo-entities as pivots and construct pseudo sentences as described in §2.2.

To efficiently find the neighboring entities from a pivot sorted by distances, we leverage the Geohash algorithm introduced by Gustavo Niemeyer in 2008. We first generate the Geohash string for geo-entities on OpenStreetMap, which encodes a location into a string of a maximum length of 20 with base 32. For example, a city ‘Madelia’ with the geolocation (44.0508° N, 94.4183° W) has a Geohash string of ‘9zufe7nwjefpjksty0m8’. The length of the hash string is associated with the size of the geographic area that the string represents. The first character divides the entire Earth’s surface into 32 large grids, and the second character further divides one large grid into 32 grids. Thus, by comparing the leading
characters of two hash strings, we can estimate the spatial proximity of two locations without calculating the exact distance between them. We use the hash strings to first filter out the geo-entities guaranteed to be outside the spatial context radius and keep the neighbors within the nearest nine hash grids for distance calculation. The use of Geohash reduces the computation time when constructing the pseudo sentences. Fig. 3 shows an example of Geohash representations. Given the pivot ‘Kenneth H Keller Hall’, we can calculate its hash string of different lengths to quickly retrieve the neighboring entities within a desired range (i.e., the green box).

This way, we generate 69,168 and 148,811 pseudo sentences in London and California, respectively, leading to a pretraining corpus of around 150M words. We use all of them for pretraining SPA/BERT with MLM and MEP.

2.4 Inference

The pretrained SPA/BERT can already generate spatially varying contextualized representations for geo-entities. The representations support various downstream applications, including contextualized geo-entity classification and similarity-based inference tasks, such as geo-entity typing and linking.

Contextualized Geo-entity Classification Classification of a geo-entity is crucial for recognizing and resolving geospatial concepts and the grounding of geo-entities in various data sources. Here, we use geo-entity typing as an example task to demonstrate the effectiveness of contextualized geo-entity representations. In this task, we aim to predict the geo-entity’s semantic type (e.g., transportation and healthcare). We stack a softmax prediction head on top of the final-layer hidden states (i.e., geo-entity embeddings) to perform the classification.

Similarity-based Inference Integrating multi-source geographic datasets often involves finding the top $K$ nearest geo-entities in some representation space. One example task, which we refer to as geo-entity linking, is to link geo-entities from a geographic information system (GIS) oriented dataset to graph-based knowledge bases (KBs). In such a task, we can directly use the contextualized geo-entity embeddings from SPA/BERT and calculate the embedding similarity based on some metrics, such as the cosine distance, to match the corresponding geo-entities from separate data sources.

3 Experiments

We evaluate SPA/BERT on supervised geo-entity typing and unsupervised geo-entity linking tasks (§3.1-§3.3). We also investigate how SPA/BERT performs under various common characteristics of geographic data sources (e.g., entity omission from cartographic generalization) and how critical technical components and their parameters affect the performance (§3.4).

3.1 Experimental Setup

Datasets Open Street Map (OSM) is a large-scale geographic database, and it is widely used in many popular map-related services. For supervised geo-entity typing, we randomly select 25,872 and 7,726 pivot geo-entities together with their neighboring entities from point features of nine semantic types on OSM in the State of California and the City of London, respectively (Tab. 1). Each geo-entity is associated with a name and its geocoordinates. We use 80% of the data in both regions for training and 20% for testing. The task is to predict the OSM semantic type for each geo-entity. (§3.2).

For unsupervised geo-entity linking, we randomly select 14 scanned historical maps in California from the United States Geological Survey (USGS) Historical Topographic Maps Collection. We manually transcribe text labels in these maps to generate 154 geo-entities, each containing a name (from the text label) and the image coordinates (text label’s pixel location in the scanned map). The task is to find the corresponding Wikidata geo-entity for each USGS geo-entity with only their spatial relations from pixel locations. Note that the geocoordinates of geo-entities in the scanned maps are unknown. We select 15K Wikidata geo-entities having an identical name to one of the USGS geo-entities and then randomly add another 30K Wikidata geo-entities to construct the final Wikidata dataset of 45K geo-entities. For the ground truth, we manually identify the matched Wikidata geo-entity for each USGS geo-entity.

For geo-entity linking task, the annotation details are described below. We hire three undergraduate students (not the co-authors) to transcribe text labels on the USGS maps. They draw bounding boxes/polygons around text labels and transcribe

---

3Entities with the coordinate location attribute (https://www.wikidata.org/wiki/Property:P625)

4Two geo-entities can have the same name (e.g., Los Angeles, CA vs. Los Angeles, TX)
Table 1: OSM Geo-entity typing dataset statistics. The first column shows the nine OSM semantic types, following by the numbers of samples for each class in California and London.

<table>
<thead>
<tr>
<th>Classes</th>
<th>California</th>
<th>London</th>
</tr>
</thead>
<tbody>
<tr>
<td>Education</td>
<td>6,222</td>
<td>618</td>
</tr>
<tr>
<td>Entertainment_Arts_Culture</td>
<td>1,380</td>
<td>601</td>
</tr>
<tr>
<td>Facilities</td>
<td>574</td>
<td>179</td>
</tr>
<tr>
<td>Financial</td>
<td>2,590</td>
<td>769</td>
</tr>
<tr>
<td>Healthcare</td>
<td>3,779</td>
<td>1,779</td>
</tr>
<tr>
<td>Public_Service</td>
<td>2,658</td>
<td>393</td>
</tr>
<tr>
<td>Sustenance</td>
<td>4,276</td>
<td>1,693</td>
</tr>
<tr>
<td>Transportation</td>
<td>4,226</td>
<td>1,618</td>
</tr>
<tr>
<td>Waste_Management</td>
<td>167</td>
<td>76</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>25,872</strong></td>
<td><strong>7,726</strong></td>
</tr>
</tbody>
</table>

The text string to indicate geo-entity names and their locations. The results are verified and corrected by another undergraduate student and one Ph.D. student. Then, we run an automatic script to find the potential corresponding geo-entity in Wikidata by matching their names. The collected Wikidata URIs are noisy, and we have one Ph.D. student verify the URIs by marking them as True or False linking. The verification results are randomly further reviewed by a senior digital humanities scholar specialized in the representation and reception of historical places.

**Model Configuration** We create two variants of SPABERT, namely SPABERT\textsubscript{base} and SPABERT\textsubscript{large} with weights and tokenizers initialized from the uncased versions of the BERT\textsubscript{base} and BERT\textsubscript{large}, respectively. During pretraining, we mix the masked instances for MLM and MEP tasks in the same ratio. We primarily use the same optimization parameters as in (Devlin et al., 2019) and train the model with the AdamW optimizer. We save the model checkpoints every 2K iterations. The learning rate is $5 \times 10^{-5}$ for SPABERT\textsubscript{Base} and $1 \times 10^{-6}$ for SPABERT\textsubscript{Large} for stability. Distance normalization factor $Z$ is 0.0001, and distance separator DSEP is 20. The batch size is 12 to fit in one NVIDIA RTX A5000 GPU with 24GB memory.

**Model Comparison** We compare SPABERT with several strong PLMs, including BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), SpanBERT (Joshi et al., 2020), LUKE (Yamada et al., 2020), and SimCSE (Gao et al., 2021). For all these models, we use the pretrained weights with the uncased version. For SimCSE, we use both the BERT and RoBERTa versions with the weights obtained from unsupervised training. We further finetune these baseline LMs end-to-end on the same pseudo-sentence dataset as SPABERT (Tab. 1) with a softmax prediction head appended after the baseline model. Note that these baseline models only use the token and position embeddings but not the spatial coordinate embeddings.

**Result Analysis** Tab. 2 summarizes the geo-entity typing results on the OSM test set with finetuned baselines and SPABERT. The last column shows the micro-F1 scores (per-instance average). The remaining columns are the F1 scores for individual classes. For all baseline models, BERT\textsubscript{Base} and SimCSE\textsubscript{BERT-Large} (BERT\textsubscript{Large} as a close second) have the highest F1 in the base and large groups, respectively. In addition, SPABERT shows the best performance over the existing context-aware LMs for both base and large groups. We hypothesize that SPABERT’s advantage comes from 1) the pretraining tasks, especially MEP, and 2) the spatial hash grids. Note that during training, the model does not use the neighboring geo-entities’ OSM semantic types. We report the F1 score for each class (i.e., an OSM semantic type) and the micro-F1 for all samples in the test set.
coordinate embeddings. Additionally, compared to its backbone, BERT, SPABERT shows performance improvement in almost all classes for both base and large models. The results demonstrate that SPABERT can effectively contextualize geo-entities with their spatial context.

We can also observe that the Financial class has high F1 scores for all models. The reason is that when entity names are strongly associated with the semantic type (e.g., U.S. Bank), pretrained LMs could produce meaningful entity representation even without the spatial context. However, the typing performance can still be further improved after encoding the spatial context with SPABERT. Fig. 4 visualizes the geo-entity features from SPABERT Base. It shows that Financial and Public_Service have the most separable features from other classes. The non-separable region in the middle is mainly due to the similar spatial context and semantics of geo-entities (e.g., “nursing home” can be close to both Facilities and Healthcare).

### 3.3 Unsupervised Geo-Entity Linking

**Task Description** We define the problem as follows. For a query set \( Q = \{(q_i, SC(q_i))\}_{i=1}^{|Q|}\), the goal is to find the corresponding geo-entity for each \( q_i \) in the candidate set \( C = \{c_i, SC(c_i)\}_{i=1}^{|C|} \) (typically \( |C| \gg |Q| \)). Here, \( Q \) and \( C \) are the USGS and Wikidata geo-entities (§3.1). \( SC(q_i) \) contains \( q_i \)'s nearest 20 geo-entities, and \( SC(c_i) \) includes \( c_i \)'s neighboring geo-entities within a 5km radius. For this task, we use the evaluation metrics of Recall@K and Mean-Reciprocal-Rank (MRR).

**Model Comparison** We use the same baseline models as in the typing task without finetuning.

### Result Analysis

Tab. 3 shows the geo-entity linking results. Large models perform better than their base counterparts for both baseline models and SPABERT. In particular, among the baseline models, SimCSE BERT has the highest score for MRR and R@1. SPABERT has significantly higher R@5 and R@10 than SimCSE. Also, SPABERT outperforms all baselines on MRR, R@5, and R@10. For both base and large versions, SPABERT outperforms its backbone model BERT. The difference between BERT and SPABERT shows the effectiveness of the spatial coordinate embeddings and domain adaptation using MLM and the proposed MEP. Also, SpanBERT shows the worst performance among all models. This could be that SpanBERT utilizes masked random spans in pretraining, which does not work well for pseudo sentences of geo-entity names. In contrast, SPABERT’s MEP specifically masks tokens from the same

---

Table 2: Geo-entity typing with the state-of-the-art LMs and SPABERT. Column names are the OSM classes. Bold numbers are the highest scores in each column. Underlined numbers are the highest scores among baselines.

<table>
<thead>
<tr>
<th>Model</th>
<th>MRR</th>
<th>R@1</th>
<th>R@5</th>
<th>R@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT Base</td>
<td>.674</td>
<td>.624</td>
<td>.763</td>
<td>.929</td>
</tr>
<tr>
<td>RoBERTa Base</td>
<td>.626</td>
<td>.672</td>
<td>.605</td>
<td>.951</td>
</tr>
<tr>
<td>SpanBERT Base</td>
<td>.613</td>
<td>.589</td>
<td>.608</td>
<td>.916</td>
</tr>
<tr>
<td>LUKE Base</td>
<td>.648</td>
<td>.608</td>
<td>.598</td>
<td>.945</td>
</tr>
<tr>
<td>SimCSE BERT Base</td>
<td>.623</td>
<td>.590</td>
<td>.504</td>
<td>.925</td>
</tr>
<tr>
<td>SPABERT Base</td>
<td>.674</td>
<td>.653</td>
<td>.680</td>
<td>.959</td>
</tr>
</tbody>
</table>

Table 3: Geo-entity linking result. Bold and underlined numbers are the highest scores in each column and the highest scores among the baselines, respectively.
geo-entity and could effectively contextualize geo-entity names distributed in the 2D space with the spatial coordinate embeddings.

### 3.4 Spatial Context and Ablation Study

**Impact of Entity Omission for Linking** We analyze how SpaBERT handles different USGS map scales for geo-entity linking with entity omission from cartographic generalization (e.g., some entities only exist in certain scales). Here SpaBERT performs the best for the 1:125K maps (30-CA) (Tab. 4). Our hypothesis is that the 1:125K maps contain denser geo-entities than the test maps at other scales. When selecting the top K nearest neighbors for a pivot in other map scales, some neighbors could be far away and would not provide relevant context.

Since the entity omission criteria of the USGS maps are unknown and Wikidata mostly contain important landmark geo-entities, we also simulate omission using the OSM dataset by gradually removing random neighbors of a pivot entity within a fixed neighborhood. We test this scenario for geo-entity linking to see if the same pivot entity would have similar representations from decreasing neighbor densities (i.e., the original set of neighbors vs. after random omission at varying rates). We use the same evaluation metric as in §3.3, as they reflect the similarity of the learned geo-entity representation (Fig. 5). The linking scores decrease with additional entities removed, indicating the similarity between the same pivot entity’s representations learned from the original data and the omitted data declines when omission percentage becomes larger. The elbow point is at about 30%, showing that SpaBERT is reasonably robust if the percentage of the removed neighbors is within 30%.

**Impact of Pseudo Sentence Length** We analyze the impact of pseudo sentence length on geo-entity typing by varying the number of neighboring entities. Tab. 5 shows the results of SpaBERT_{Base} and SpaBERT_{Large} with an increasing number of neighbors. We observe that as #Neighbor increases, the micro-F1 score increases accordingly, as expected. The increment is more evident at the beginning, illustrating that the spatial context helps contextualize the pivot entity, especially in a local neighborhood. The benefit of adding additional neighboring entities decreases as distance increases beyond a certain point. This conforms to the well-studied spatial autocorrelation.

**Effect of Spatial Coordinate Embedding** We train two additional SpaBERT variants that do not include the spatial coordinate embedding during MLM and MEP pretraining and use them for geo-entity linking. For MRR, SpaBERT_{Base} and SpaBERT_{Large} drop from 0.515 to 0.458 and from 0.537 to 0.478 compared to their original SpaBERT version. Also, the variants have lower recall scores compared to their original SpaBERT version. The most significant drop in the recall is on R@1 from 0.383 to 0.283 for SpaBERT_{Large}. The results demonstrate the effectiveness of the spatial coordinate embedding.

### 4 Related Work

**Pretrained Language Models** PLMs have been the dominant paradigm for language representation. Following the success of MLMs (Devlin et al., 2019; Liu et al., 2019) and autoregressive PLMs...
more recent work has extended the pretraining process with more tasks or learning objectives for span prediction (Joshi et al., 2020), cross-encoders (Reimers and Gurevych, 2019), contrastive learning (Gao et al., 2021), and massively multi-task learning (Raffel et al., 2020). To support entity representation, several approaches propose to perform mention detection (Yamada et al., 2020), incorporate mention memory cells (de Jong et al., 2021), or injecting structural knowledge representations (Wang et al., 2021; Zhang et al., 2019; Peters et al., 2019; Zhou et al., 2022). Due to the large body of work, we refer readers to recent surveys summarizing this line of work (Qiu et al., 2020; Wei et al., 2021).

To extend the use of PLMs beyond language, much exploration has also been conducted to represent other modalities. For example, a significant amount of vision-language models (Kim et al., 2021; Zhai et al., 2022; Li et al., 2020a; Lu et al., 2019) have been developed by jointly pretraining on co-occurring vision and language corpora, and are in the support of grounding (Zhang et al., 2020; Chen et al., 2020), generation (Cho et al., 2021; Yu et al., 2022) and retrieval tasks (Zhang et al., 2021; Li et al., 2020b) on the vision modality. Other PLMs capture semi-structure tabular data by linearizing table cells (Herzig et al., 2020; Yin et al., 2020; Iida et al., 2021) or incorporating structural prior in the attention mechanism (Wang et al., 2022; Trabelsi et al., 2022; Eisenschlos et al., 2021). To the best of our knowledge, none of the prior studies have extended pretrained LMs to geo-entity representation, nor do they support a suitable mechanism to capture the geo-entities’ spatial relations, which do not have structural prior. This is exactly the focus of our work.

Domain-specific Language Modeling Another line of studies has been conducted to adapt PLMs to specific domains typically by pretraining on domain-specific corpora. For example, in the biomedicine domain, a series of models (Lee et al., 2020; Peng et al., 2019; Alsentzer et al., 2019; Phan et al., 2021) have been developed by training PLMs on corpora derived from PubMed. Similarly, PLMs have been trained on corpora specific to software engineering (Tabassum et al., 2020), finance (Liu et al., 2021), and proteomics (Zhou et al., 2020) domains. In this context, SPABERT represents a pilot study in the geographical domain by allowing the PLM to learn from spatially distributed text.

5 Conclusion
This paper presented SPABERT, a language model trained on geographic datasets for contextualizing geo-entities. SPABERT utilizes a novel spatial coordinate embedding mechanism to capture spatial relations between 2D geo-entities and linearizes the geo-entities into 1D sequences, compatible with the BERT-family structures. The experiments show that the general-purpose representations learned from SPABERT achieve better or competitive results on the geo-entity typing and geo-entity linking tasks compared to SOTA pretrained LMs. We plan to evaluate SPABERT on additional related tasks, such as geo-entity to natural language grounding. Also, we plan to extend SPABERT to support other geo-entity geometry types, including lines and polygons.

Limitations
The current model design only considers points but not polygon and line geometries, which could also help provide meaningful spatial relations for contextualizing a geo-entity. Training of SPABERT also requires considerable GPU resources which might produce environmental impacts.

Ethical Consideration
SPABERT was evaluated on the English-spoken regions, so the model and results could have a bias towards these regions and their commonly used languages. Replacing the backbone of SPAERT with a multi-lingual model and training SPABERT with diverse regions could mitigate the bias.

Acknowledgement
We thank the reviewers for their insightful comments and suggestions. We thank Dr. Valeria Vitale for her valuable input. This material is based upon work supported in part by NVIDIA Corporation, the National Endowment for the Humanities under Award No. HC-278125-21 and Council Reference AH/V009400/1, the National Science Foundation of United States Grant IIS 2105329, a Cisco Faculty Research Award (72953213), and the University of Minnesota, Computer Science & Engineering Faculty startup funds.
References


Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Long, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zaraana Parekh, Xin Li, Han Zhang, Jason Baldrige, and Yonghui Wu. 2022. Scaling Autoregressive Models for Content-Rich Text-to-Image Generation.


